Background Check: A general technique to build more reliable and versatile classifiers

Miquel Perello-Nieto[1†], Telmo M. Silva Filho[1‡], Meelis Kull[†] and Peter Flach[†]
[†]Department of Computer Science, University of Bristol, UK
[‡]Centro de Informatica, Universidade Federal de Pernambuco, Brazil
Email: [†]{Miquel.PerelloNieto, Meelis.Kull, Peter.Flach}@bristol.ac.uk, [‡]tmsf@cin.ufpe.br

***Abstract*—We introduce a powerful technique to make classifiers more reliable and versatile. *Background Check* equips classifiers with the ability to assess the difference of unlabelled test data from the training data. In particular, Background Check gives classifiers the capability to (i) perform cautious classification with a reject option; (ii) identify outliers; and (iii) better assess the confidence in their predictions. We derive the method from first principles and consider four particular relationships between background and foreground distributions. One of these assumes an affine relationship with two parameters, and we show how this bivariate parameter space naturally interpolates between the above capabilities. We demonstrate the versatility of the approach by comparing it experimentally with published special-purpose solutions for outlier detection and confident classification on 41 benchmark datasets. Results show that Background Check can match and in many cases surpass the performances of specialised approaches.**

I. Introduction and Motivation

While making decisions, human experts and machine learning models might face two difficult scenarios, where even vast prior knowledge does not allow them to make *confident* decisions. The first scenario consists of cases which are known from experience to be *ambiguous*. For example, a bank manager who wants to decide if a loan application should be accepted, compares the applicant's supplied information to the bank's database, but this might not give a clear prediction of loan repayment chance. In this situation, the manager could do a background check (criminal and credit history) before making a decision. The second scenario happens when new or *unknown* cases appear. For example: a scientist analysing data from an experiment might come across some unusual values. Further verification could be done to either identify the new case as an unexpected relevant discovery or dismiss it as a measuring defect. In both scenarios, a human expert's first answer would be "I don't know" or "I am not *confident* enough", but the reasons for this low confidence are different. The vast majority of machine learning models are not trained to give such an answer, while the ones that try to do it focus only on one of these scenarios. We argue that a unified view is called for, with the different scenarios being special cases.

The key idea of this paper is to equip classifiers with the ability to assess the difference of test data from the labelled *foreground data* on which the classifier was trained. This ability arises from assumptions about the nature of *background data* based on available knowledge about *foreground data* and the task to be performed. Since we use *background* information to perform these tasks, our unified technique is called *Background Check* (BC). In test data we expect some instances to be from *background*, which we consider as an additional, novel class $k+1$, even if it may actually be very heterogeneous and not form a class as such. BC learns to estimate $(k+1)$-class probabilities from k-class training data, where the extra class represents *background* data. We propose that the addition of the extra class provides the classifier with very useful capabilities including the following:

1) **perform cautious classification with reject option:** The goal of this task is to build classifiers that can refrain from producing an output for ambiguous instances, i.e. instances with posterior probabilities for all classes lower than a certain threshold. Our approach addresses this by assuming that *foreground* and *background* data have similar distributions, with the *background* density playing a key role in the rejection rate.
2) **identify outliers:** Here, special techniques are employed, capable of detecting when a new instance differs from known classes. If we assume that outliers emerge from regions where *foreground* data are less dense than *background* data, we are able to reveal whether an instance is an outlier from its posterior probabilities.
3) **better assess the confidence in its predictions:** Imagine a binary classification problem where both classes are equiprobable and the likelihoods of two samples on class C_1 are $p(x_1|C_1) = 0.9$ and $p(x_2|C_1) = 0.1$, while for class C_2 both likelihoods are zero. After applying Bayes' rule, we get the same posterior probabilities for both examples: 1 for C_1 and 0 for C_2. Therefore, both samples have the same predictions. However, the model should be more confident for sample x_1 than for x_2, as the likelihood of the first sample on class C_1 is higher. By providing probabilities for *foreground* and *background* data, our approach gives a solution to this problem.

BC is general in the sense that it works as a wrapper and hence does not assume any particular model class. One side effect of our approach is that we are able to extract posterior probabilities from one-class classifiers, such as the well-known one-class support vector machine (OCSVM) [1], for which, to the best of our knowledge, there was no way of doing so.

[1]The first two authors contributed equally to this work.

We organised this paper as follows: Section II explains BC in detail; in Section III, we present some applications of our approach and discuss previous works from the literature which investigated these tasks, while showing how to use BC to solve them; Section IV describes our experimental analysis, where we compare BC against methods that were specifically designed for each task; and Section V discusses the main findings and concludes. Supplementary material[2] is available containing proofs, algorithms and more detailed results.

II. The Background Check Method

In this paper any instance x necessarily belongs to either foreground or background and not to both at the same time. If it belongs to the foreground, then it belongs to exactly one of the $k \geq 1$ classes. Formally, we represent the class of x with a single label $y \in \{1,\ldots,k,k{+}1\}$, where the values $1,\ldots,k$ represent the k foreground classes and the value $k{+}1$ represents the background. The background class is special as we have very little or no training data from that class.

To solve any of the tasks listed in the introduction we will present methods to estimate the $(k{+}1)$-class posterior probability distribution $p(Y|X{=}x)$, where (X,Y) denotes a randomly drawn (i.i.d.) test instance with known features $X{=}x$ but unknown label $Y \in \{1,\ldots,k{+}1\}$. For notational convenience we denote the events $Y{=}1, \ldots, Y{=}k$ and $Y{=}k{+}1$ as $\mathsf{f}_1,\ldots,\mathsf{f}_k$ and b, respectively, and the event $Y \in \{1,\ldots,k\}$ as f. Also, by a slight abuse of notation we write simply x to refer to the event $X{=}x$. In this new notation, our main task is to estimate the probabilities $p(\mathsf{f}_1|x),\ldots,p(\mathsf{f}_k|x)$ and $p(\mathsf{b}|x)$.

What makes our task special is the lack or shortage of data from the background class. In such cases standard methods fail in the sense that the probability of background will be zero or very close to zero for any instance. However, standard methods still allow us to estimate the probabilities for the foreground classes, conditioned on the assumption that the instance is from the foreground. That is, we can apply any multi-class probability estimator algorithm on the training data from the foreground classes to estimate the probabilities $p(\mathsf{f}_1|\mathsf{f},x),\ldots,p(\mathsf{f}_k|\mathsf{f},x)$. We refer to these probabilities as the *class posterior probabilities within foreground*.

A. The Familiarity Factor

The class posterior probabilities within foreground do not provide any information to estimate the foreground and background probabilities $p(\mathsf{f}|x)$ and $p(\mathsf{b}|x)$ which are part of our main estimation task. That is, even for fixed probabilities within foreground the value of $p(\mathsf{f}|x) = 1 - p(\mathsf{b}|x)$ can still have any value from 0 to 1. In particular, $p(\mathsf{f}|x) = 1$ and $p(\mathsf{b}|x) = 0$ refers to absolute certainty that instance x belongs to the foreground, i.e., it belongs to one of our familiar classes with sufficient training data. Conversely, $p(\mathsf{f}|x) = 0$ and $p(\mathsf{b}|x) = 1$ refers to complete certainty that the instance is in an unfamiliar region of the instance space, making us fully confident that it does not belong to the foreground. This intuition justifies our following definition.

Definition 1. *For any instance x we define its* familiarity factor *as* $r(x) = p(\mathsf{f}|x)/p(\mathsf{b}|x)$.

It turns out that knowing the class posterior probabilities within the foreground together with the familiarity factor is sufficient to obtain $(k{+}1)$-class posterior probabilities.

Proposition 1.

$$p(\mathsf{b}|x) = \frac{1}{1+r(x)}, \quad p(\mathsf{f}_c|x) = \frac{p(\mathsf{f}_c|\mathsf{f},x)r(x)}{1+r(x)} \quad \textit{for } c = 1,\ldots,k$$

Proof. All proofs can be found in the supplementary material. □

Furthermore, the familiarity factor is not only sufficient but also necessary in the sense that it can be directly obtained from the $(k{+}1)$-class posterior probabilities:

$$r(x) = \frac{p(\mathsf{f}|x)}{p(\mathsf{b}|x)} = \frac{p(\mathsf{f}_1|x)+\cdots+p(\mathsf{f}_k|x)}{p(\mathsf{b}|x)}.$$

We will next study how to estimate the familiarity factor.

If despite the shortage of background data we have enough to learn a two-class probability estimator to distinguish between foreground and background, then we can calculate the familiarity factor directly from its definition $r(x) = p(\mathsf{f}|x)/p(\mathsf{b}|x)$. This also becomes possible if we have a way to generate background data synthetically. In the following we consider the case when learning a discriminatory model is infeasible. First we note that $r(x) = p(x,\mathsf{f})/p(x,\mathsf{b})$ because $p(x,\mathsf{f}) = p(\mathsf{f}|x)p(x)$ and $p(x,\mathsf{b}) = p(\mathsf{b}|x)p(x)$ where $p(x)$ is the density of test data at x. We refer to $p(x,\mathsf{f})$ and $p(x,\mathsf{b})$ as foreground and background densities, respectively. Since we are interested in the ratio of these densities, we care only about the relative densities, rather than absolute.

Definition 2. *We define the* relative foreground density $q_\mathsf{f}(x)$ *and* relative background density $q_\mathsf{b}(x)$ *as follows:*

$$q_\mathsf{f}(x) = \frac{p(x,\mathsf{f})}{\max_x p(x,\mathsf{f})}, \qquad q_\mathsf{b}(x) = \frac{p(x,\mathsf{b})}{\max_x p(x,\mathsf{f})}.$$

Note that both of these densities are taken relative to the same quantity, which is the maximal foreground density across the whole instance space. It is now easy to see that the familiarity factor can be calculated as the ratio of relative foreground and background densities: $r(x) = q_\mathsf{f}(x)/q_\mathsf{b}(x)$.

B. Estimating Relative Foreground/Background Densities

It turns out that relative foreground density estimation is equivalent to the task of estimating $p(x|\mathsf{f})$, in the sense that either of these can be directly calculated from the other.

Proposition 2.

$$q_\mathsf{f}(x) = \frac{p(x|\mathsf{f})}{\max_x p(x|\mathsf{f})}, \qquad p(x|\mathsf{f}) = \frac{q_\mathsf{f}(x)}{\int_x q_\mathsf{f}(x)dx}.$$

Therefore, one can first use any standard density estimation algorithm and then calculate the relative foreground density.

[2]Supplementary material with code, mathematical proofs and extended results is available at https://reframe.github.io/background_check/

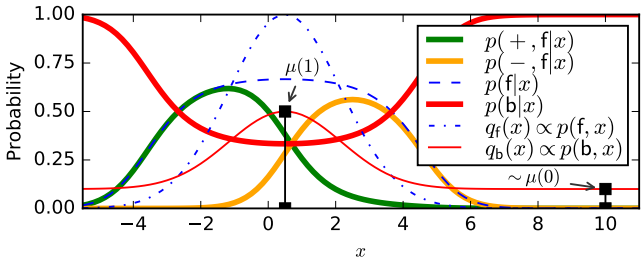

Fig. 1. Example of Background Check (BC) assuming the affine background bias. For details see text and the supplementary material.

Often we tend to know very little about the outliers or novel classes to emerge. Therefore, we need a strong inductive bias with regard to the *background* to obtain $q_\mathsf{b}(x)$. In particular, we introduce four background biases in order of increasing strengths. We hypothesise that for most practical applications the lack of information about the *background* forces the use of one of the two strongest biases.

1) Under the first, least restrictive background bias we assume that q_b is a function of q_f, i.e. $q_\mathsf{b}(x) = \mu(q_\mathsf{f}(x))$. This makes sense, as due to lack of information there are no reasons to assign different *background* densities to any two points which have equal foreground density. Under this background bias the domain knowledge must inform the choice of function $\mu : [0,1] \to [0,\infty)$. If the domain knowledge is insufficient for this, a stronger inductive bias is required.
2) Under the second, *monotonic* background bias we assume that the function μ is either monotonically increasing or monotonically decreasing. That is, when moving to a region with higher foreground density the *background* density increases, or decreases, respectively. The justification of this assumption is that due to lack of information there are no reasons to have particular points in μ as local minima or maxima.
3) Under our third, affine background bias we assume that the function μ has the form $\mu(z) = az+b$. Instead of the parametrisation with a and b we choose to parametrise by $\mu(0)$ and $\mu(1)$, the end-points of the function (see Figure 1). That is, $\mu(z) = (1-z)\mu(0) + z\mu(1)$. If the domain knowledge is not even sufficient to provide $\mu(0)$ and $\mu(1)$ then the strongest bias needs to be considered.
4) Under our fourth and strongest *constant* background bias we assume that $\mu(z) = 0.5$. This is a special case of the affine background bias with $\mu(0) = \mu(1) = 0.5$.

Figure 1 shows an example of BC with two normally distributed *foreground* classes and assuming the affine background bias, with $\mu(0) = 0.1$ and $\mu(1) = 0.5$. We can see that the background is half as dense as the foreground in the densest foreground region. Moreover, the densities of foreground and background decrease together, but the background density never becomes less than $\mu(0) = 0.1$, eventually becoming denser than the foreground. The maximum familiarity $r(x)$ occurs at $\mu(1)$, while the lowest familiarity occurs at $\mu(0)$.

In cases where we need strong background biases due to lack of data about the background we can relax the foreground density estimation. In particular, we may care more about estimating which regions have higher and which lower foreground density, but less about what the exact density values are. The advantage of this situation is that we can use one-class scoring classifiers: models which assign higher scores to regions which are inside the foreground distribution and lower scores to regions outside. All we need to do is to make sure that we monotonically transform the scores to be between 0 and 1, to obtain an approximate relative foreground density. For this there are many possible transformations and the choice is to be made based on the knowledge about the scoring classifier and the domain. For example, OCSVM [1] is expected to output positive values for *foreground* and negative values for *background* data. In this case, a sigmoid function $q_\mathsf{f}(x) = 1/(1+exp(s^* - s(x)))$ can be applied to the outputs $s(x)$ of the model, with the origin of the sigmoid fixed to s^*, which is the lowest output obtained from training data. Thus, instances that are *foreground* training data now have outputs in $[0.5, 1.0]$ and the lower bound of the output becomes 0. This meets the requirements for BC and allows us to extract posterior probabilities from OCSVM.

C. Performing Background Check

Background Check can be performed in two ways, depending on the task and on the assumed background bias. The simplest one, which we call *discriminative approach* (BCD), is built by uniformly generating artificial *background* instances around *foreground* data and training a binary discriminative model to separate them. These instances are generated in a hypercube or a hypersphere [2], such that the *background* is half as dense as $\max_x p(x,\mathsf{f})$ in every point within some bounds. This matches our fourth, constant background bias. During test time we combine the estimates given by the k-class *foreground* classifier and the *foreground* vs *background* classifier to obtain $(k+1)$-class posterior probability estimates.

The second BC method, called *familiarity approach* (BCF), is more general and works with all our background biases. First, we fit a one-class model on *foreground* data. From the scores of this model, we obtain q_f (using the sigmoid, if the model is OCSVM). Then, we use one of our background biases to obtain q_b from q_f. We can then calculate the familiarity factor $r = q_\mathsf{f}/q_\mathsf{b}$, from which we extract posterior probabilities $p(\mathsf{b}|x)$ and $p(\mathsf{f}|x)$ to obtain the $(k+1)$-class posterior probability estimates. Although BCD is simpler than BCF, the latter might be more appropriate in high-dimensional spaces, where generating artificial instances can be expensive [3].

III. Applications and Related Work

In this section we discuss how to use BC to achieve the capabilities mentioned in the Introduction.

A. Cautious classification

Training a classifier to perform cautious classification can be very useful in many areas including medical applications [4], activity recognition [5] and intrusion detection [6].

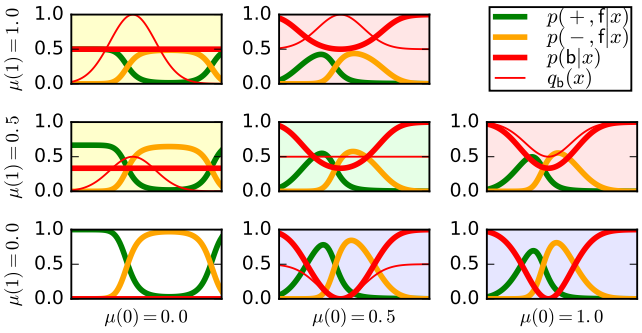

Fig. 2. Different regions on the parameter space of the Background Check assuming the affine background bias. For details see the supplementary material.

An early contribution to this task was Chow's rule [7], following which a classifier rejects an instance x if $\forall_i, p(y=i|x) < \theta$, where θ is the rejection threshold. This rule tends to reject instances close to the decision boundary, where classes have lower probabilities. Later works use either the same threshold for all classes [5], [6] or different thresholds for each class [8], [9] (which is better-suited for problems with imbalanced classes). Some papers also investigated threshold selection in cost-sensitive scenarios [6], [9].

We can mimic this cautious behaviour with BC by assuming our affine background bias, setting $\mu(0) = 0$ and varying the value of $\mu(1)$ according to the desired rejection rate. Proposition 3 offers a justification for this choice.

Proposition 3. *Given a threshold θ, if μ is the affine background bias with $\mu(0) = 0$ and $\mu(1) = \theta$, then $p(\mathrm{f}|x)$ is a monotonically decreasing function of θ of the form $p(\mathrm{f}|x) = 1/(\theta+1)$.*

As a result of Proposition 3, when $\mu(0) = 0$, foreground and background posterior probabilities are independent of the foreground density. Moreover, higher values of $\mu(1)$ lead to lower constant foreground probabilities and higher constant background posterior probabilities (as shown in the left column of Figure 2), resulting in higher rejection rates.

To perform cautious classification with BC, one must simply set $\mu(0) = 0$ and $\mu(1) = \theta$. Then, for every instance x, predict $\hat{y} = \mathrm{argmax}_i p(y=i|x)$ and reject x if $\hat{y} = k+1$.

B. Outlier detection

Outliers are instances that originate from unknown *background* regions. Outlier detection has been tackled by many different types of algorithms [10]–[13]. Among these methods, density estimators, one-class SVMs and approaches that generate artificial *background* data are of particular interest.

A good density estimator is expected to output lower values for outliers than for *foreground* data. [3] proposed a one-class classification approach that aims to improve an initial density estimator by training a discriminative model to separate training data from artificial data sampled from the generative model. If the original density estimator was poorly-fitted, the discriminative model will improve the combined performance.

[2], [14] proposed a process for building classifiers that "protect" a target class against outliers and any possible new classes by generating new artificial instances uniformly around the target class. This is equivalent to assuming our constant background bias.

[15] proposed the construction of a multi-class classifier with outlier detection as an ensemble with one one-class classifier per class. Because different one-class classifiers trained on different classes will output values in different scales, they proposed two ways of normalising these outputs, called outlier normalisation (O-norm) and target normalisation (T-norm). The difference between these approaches lies in the values given to objects that fall in the rejection boundary.

Both O-norm and T-norm are model-dependent. Moreover, O-norm and T-norm might also result in one class dominating the others if its scores are distributed with higher variance. Similarly to their approach, we can build multi-class classifiers with outlier detection with one BC per class. Since BC always outputs probabilities $p(\mathrm{f}|x)$, regardless of the base model, the outputs do not need to be normalised in a model-dependent way, as in [15], while also mitigating the problem of one class dominating the other ones.

In a general multi-class outlier detection scenario, there are two ways of applying BC. The first one, represented by the central plot in Figure 2, assumes the constant background bias and employs our discriminative approach (BCD). The second approach, represented by the bottom row in Figure 2, considers that outliers are denser in regions where *foreground* data are less dense and vice versa and uses our familiarity approach (BCF) by assuming the affine background bias, with $\mu(1) = 0$ and $\mu(0) > 0$. Both BCD and BCF will produce $(k+1)$-class posterior probability estimates and predict $\hat{y} = \mathrm{argmax}_i p(y=i|x)$ for every test instance x, marking x as outlier if $\hat{y} = k+1$.

C. Classification with confidence

The benefits of considering the confidence of a classifier's predictions have been discussed before in the machine learning literature [16]–[19]. Most approaches treat a classifier's outputs or class-conditional probability estimates as confidence. [19] discussed the importance of confidence in weighted voting schemes in ensemble learning. Their approach associates a confidence level to the prediction given by each classifier of the ensemble for a given example. The weights of the classifiers are trained by minimising a cost function that is akin to training a maximum margin SVM on the classifiers' predictions multiplied by their respective confidence values. After training the weights, the ensemble is pruned, by sorting the classifiers according to weight and keeping the classifiers that form the sub-ensemble with maximum training accuracy.

Treating class probability outputs of a classifier as confidence has a drawback. As explained in Section I, due to the normalisation made by Bayes' rule, class-conditional probabilities might not be well-suited for some situations, such as instances that come from sparse regions of *foreground* data. To solve this problem, given any of our background biases, BC provides us with *foreground* probabilities $p(\mathrm{f}|x)$, which

we name *confidence*, and $(k+1)$-class conditional probabilities, which we call *confident probabilities*. The chosen background bias depends on which type of confidence will be evaluated, that is, confidence can represent instance ambiguity (cautious classification), unfamiliarity (outlier detection), or both.

In the particular application of ensemble learning proposed by [19] every classifier of the ensemble is built on a subset of the training data. Hence, even if no new classes are expected during deployment, these models should not be overconfident about instances that come from regions of the feature space on which they were not trained. We therefore expect that our confident probabilities should increase the ensemble's performance. Here we want to evaluate each classifier's confidence in a general way, thus, for this task we assume our constant background bias, setting $\mu(0) = \mu(1) = 0.5$. Furthermore, we hypothesise that the average confidence given by BC for the training data of a classifier is enough to know how important its predictions should be for the ensemble. Therefore, we set the weight of a classifier as its average confidence.

IV. Empirical Evaluation

The main point of this paper is to present an approach general enough to be applied to various machine learning tasks that were previously considered unrelated – or weakly related – and for which many specialised solutions were pursued, as discussed in Section III. In this section we will empirically analyse the usefulness of BC for solving these tasks, except cautious classification, which we theoretically proved to be a built-in capability of BC. For all experiments, we chose to use our familiarity approach (BCF), due to the potentially high training cost of generating artificial instances for some of the datasets. Therefore, for simplicity we will refer to the chosen approach simply as BC for the remainder of this section.

In order to demonstrate the versatility of BC we selected 41 datasets from UCI [20]. Half of them have been used previously in publications [3], [8], [15], [19] which we cite and/or compare against. Because of our interest in multiclass classification problems we selected 20 additional datasets with more than 3 classes. We preprocessed and standardised all datasets. Nominal features were transformed into numerical values. If the number of instances with missing values was less than 25% of the dataset, we removed those instances. Otherwise we kept them to avoid discarding too much information, but substituted the missing values by the mean of their corresponding feature. Datasets with more than 30 000 instances were reduced to 10% of their original size. Finally all features were standardised with mean zero and variance one. More details of the preprocessing can be found in the supplementary material.

A. Outlier detection

For this application, we chose to compare our method with the multiclass approach proposed by [15] and discussed in Section III-B. To compare O-norm and T-norm with BC, we selected the Naive Parzen density estimator, because it was one of the overall best methods in their analysis. Following their experimental setup, we selected rejection thresholds such that a 10% rejection rate was achieved for the training data of each class. To simulate the emergence of outliers during test time, we generated artificial instances from a Gaussian with four times the covariance of test data, totalling 50% of the number of test instances. Since [15] did not report which values were used for the bandwidth parameter of the Gaussian kernel used by Naive Parzen, we selected one value (0.05) for all experiments. Experiments were run on 20 times 5-fold cross-validation. For each dataset, we ranked the methods by their mean accuracies. Table I presents the results measured by accuracy and ranking order. For full details about the results, please check the supplementary material.

TABLE I
Mean accuracies and ranking over 41 datasets in the outlier detection experiments.

	BC	O-norm	T-norm
Ranked 1	**19**	11	11
Ranked 2	7	11	**19**
Ranked 2.5	0	4	4
Ranked 3	15	15	7
Mean acc.(rank)	**74.32 (1.90)**	70.95 (2.14)	72.59 (1.95)

We note that BC had better average accuracy in most datasets with 30 or more features. Moreover, O-norm seemed to perform better on the datasets with the largest number of samples, while BC performed better on small datasets. To evaluate the significance of these results we performed Friedman's test, which did not reject the null hypothesis. While O-norm and T-norm are specially designed solutions that need to be adapted for different density estimators, our method is general enough to solve the same task without the necessity of designing specific solutions.

B. Classification with confidence

We show the usefulness of the confident predictions given by BC by applying our approach to the task of confident weighted-voting ensembles proposed by [19], as explained in Section III-C. We ran 20 times 5-fold cross-validation for each dataset. For each experiment, we trained an ensemble of 100 classifiers. For binary datasets, each classifier was a linear SVM, trained on a subset with 75% of the training data selected by bootstrapping. For multiclass datasets, each classifier in the ensemble was composed of $k(k-1)/2$ pairwise linear SVMs and output the most predicted class among its pairwise components. The confidence of this prediction was the minimum output given by the pairwise components trained on the winning class.

For our approach, for each classifier in the ensemble, we trained a BC model following the assumptions we proposed in Section III-C. For the confidence values, we used the confident probabilities obtained from the BC model. The chosen one-class model for BC was OCSVM [1], as implemented by Python's scikit-learn library [21], keeping the default parameter values, except for parameter ν, which was set to 0.1.

TABLE II
MEAN ACCURACY AND LOG-LOSS OVER 41 DATASETS FOR THE CLASSIFICATION WITH CONFIDENCE EXPERIMENTS. THE RESULTS ARE SIGNIFICANTLY BETTER AT $p < 0.001$ ACCORDING TO A WILCOXON SIGNED RANK-SUM TEST. FULL RESULTS IN THE SUPPLEMENTARY MATERIAL.

	Accuracy		Log-loss	
method	EP-CC	BC	EP-CC	BC
Mean	81.54	**82.27**	3.42	**2.98**

Each ensemble outputs a matrix with the weighted votes for each class and for each instance, from which normalised class-conditional probabilities were obtained. Table II presents the final results in terms of mean accuracy and log-loss averaged for 100 results from the 41 datasets. To verify the statistical significance of these results, we performed a Wilcoxon signed rank-sum test per dataset. BC achieved higher accuracy for 32 datasets (26 statistically significant with $p < 0.05$), while EP-CC achieved better results for 9 datasets (6 statistically significant). Furthermore, BC showed lower log-loss in general, outperforming EP-CC for a total of 36 datasets (34 statistically significant). On the other hand, EP-CC had lower log-loss for 6 datasets (2 statistically significant).

V. DISCUSSION AND CONCLUSIONS

Our experimental analysis with a large number of datasets proved the usefulness of BC as a general technique, capable of performing different tasks, such as cautious classification, outlier detection and classification with confidence, for which only special-purpose methods existed.

We proved theoretically that BC is naturally equipped with the capability to perform cautious classification. As for outlier detection, although BC performed better on average than two specialised model-dependent methods, the difference was not statistically significant, according to Friedman's test. Nevertheless, there is still merit for BC on this task, because it is a simple general technique that does not depend on the base model. For the last task, we showed that our confident probabilities contribute to increasing the ensemble's accuracy, while also extracting better probability estimations from it (as evidenced by the log-loss). Due to the unified solution provided by BC, these tasks can be solved simultaneously by the same model, depending on the chosen *background* bias.

Future works include the investigation of BC under a cost perspective, and the cost surfaces that will result from the addition of the *background* class, and possible new applications for BC, such as detection of novel classes, where instances labelled as background can be further background checked in search of emerging patterns.

ACKNOWLEDGEMENTS

This work was supported by the Reframe project funded by CHIST-ERA and EPSRC under grant EP/K018728. TSF was financially supported by CNPq (Brazilian National Council for Scientific and Technological Development).

REFERENCES

[1] B. Schölkopf, R. C. Williamson, A. J. Smola, J. Shawe-Taylor, and J. C. Platt, "Support vector method for novelty detection." *NIPS*, vol. 12, pp. 582–588, 1999.

[2] T. C. W. Landgrebe, D. M. J. Tax, P. Paclk, R. P. W. Duin, and C. Andrew, "A combining strategy for ill-defined problems," *Fifteenth Ann. Sympos. of the Pattern Recognition Association of South Africa*, p. 5762, November 2004.

[3] K. Hempstalk, E. Frank, and I. H. Witten, *Machine Learning and Knowledge Discovery in Databases: European Conference, ECML PKDD 2008, Antwerp, Belgium, September 15-19, 2008, Proceedings, Part I*. Springer Berlin Heidelberg, 2008, ch. One-Class Classification by Combining Density and Class Probability Estimation, pp. 505–519.

[4] B. Hanczar and A. Bar-Hen, "Controlling the cost of prediction in using a cascade of reject classifiers for personalized medicine," in *Proc. of the 7th International Conference on Bioinformatics Models, Methods and Algorithms (BIOINFORMATICS 2016)*, Rome, Italy, 2016, pp. 42—-50.

[5] V. Carletti, P. Foggia, G. Percannella, A. Saggese, and M. Vento, "Recognition of human actions from RGB-D videos using a reject option," in *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, vol. 8158 LNCS. Springer Berlin Heidelberg, 2013, pp. 436–445.

[6] T. Pietraszek, "Classification of intrusion detection alerts using abstaining classifiers," *Intell. Data Anal.*, vol. 11, no. 3, pp. 293–316, Aug. 2007.

[7] C. Chow, "On optimum recognition error and reject tradeoff," *IEEE Transactions on Information Theory*, vol. 16, no. 1, pp. 41–46, Jan 1970.

[8] C. Ferri and J. Hernández-Orallo, "Cautious classifiers." *Proceedings of ROC Analysis in Artificial Intelligence, 1st International Workshop (ROCAI–2004)*, vol. 4, pp. 27–36, 2004.

[9] X. Zhang and B.-G. Hu, "A new strategy of cost-free learning in the class imbalance problem," *Knowledge and Data Engineering, IEEE Transactions on*, vol. 26, no. 12, pp. 2872–2885, 2014.

[10] M. Markou and S. Singh, "Novelty detection: a review-part 1: statistical approaches," *Signal Processing*, vol. 83, no. 12, pp. 2481–2497, Dec 2003.

[11] ——, "Novelty detection: a review-part 2:," *Signal Processing*, vol. 83, no. 12, pp. 2499–2521, Dec 2003.

[12] V. Chandola, A. Banerjee, and V. Kumar, "Anomaly detection," *ACM Computing Surveys*, vol. 41, no. 3, pp. 1–58, Jul 2009.

[13] M. A. Pimentel, D. A. Clifton, L. Clifton, and L. Tarassenko, "A review of novelty detection," *Signal Processing*, vol. 99, pp. 215–249, Jun 2014.

[14] T. C. Landgrebe, D. M. Tax, P. Paclík, and R. P. Duin, "The interaction between classification and reject performance for distance-based reject-option classifiers," *Pattern Recognition Letters*, vol. 27, no. 8, pp. 908–917, Jun 2006.

[15] D. Tax and R. Duin, "Growing a multi-class classifier with a reject option," *Pattern Recognition Letters*, vol. 29, no. 10, pp. 1565–1570, Jul 2008.

[16] N. Toth and B. Pataki, "On classification confidence and ranking using decision trees," in *2007 11th International Conference on Intelligent Engineering Systems*, June 2007, pp. 133–138.

[17] Q. Zhou, Y. Zhang, and X. Hu, "An ensemble method based on confidence probability for multi-domain sentiment classification," in *Lecture Notes in Computer Science*, vol. 7389 LNCS. Springer Berlin Heidelberg, 2012, pp. 214–220.

[18] L. Ma, X. Liu, L. Song, Y. Liu, C. Zhou, X. Zhao, and Y. Zhao, "A new classifier fusion method based on confusion matrix and classification confidence for recognizing common CT imaging signs of lung diseases," in *Medical Imaging*, S. Aylward and L. M. Hadjiiski, Eds., vol. 9035. International Society for Optics and Photonics, Mar 2014, pp. 90 351H–90 351H–6.

[19] L. Li, Q. Hu, X. Wu, and D. Yu, "Exploration of classification confidence in ensemble learning," *Pattern Recognition*, vol. 47, no. 9, pp. 3120 – 3131, 2014.

[20] M. Lichman, "UCI machine learning repository," 2013. [Online]. Available: http://archive.ics.uci.edu/ml

[21] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay, "Scikit-learn: Machine learning in Python," *Journal of Machine Learning Research*, vol. 12, pp. 2825–2830, 2011.